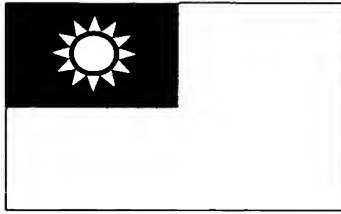


LADAS & PARRY
5670 Wilshire Boulevard
Suite 2100
Los Angeles, CA 90036
Telephone: (323) 934-2300
Telefax: (323) 934-0202



中華民國經濟部智慧財產局

INTELLECTUAL PROPERTY OFFICE
MINISTRY OF ECONOMIC AFFAIRS
REPUBLIC OF CHINA

茲證明所附文件，係本局存檔中原申請案的副本，正確無訛，
其申請資料如下：

This is to certify that annexed is a true copy from the records of this
office of the application as originally filed which is identified hereunder:

申請日：西元 2003 年 04 月 15 日
Application Date

申請案號：092108725
Application No.

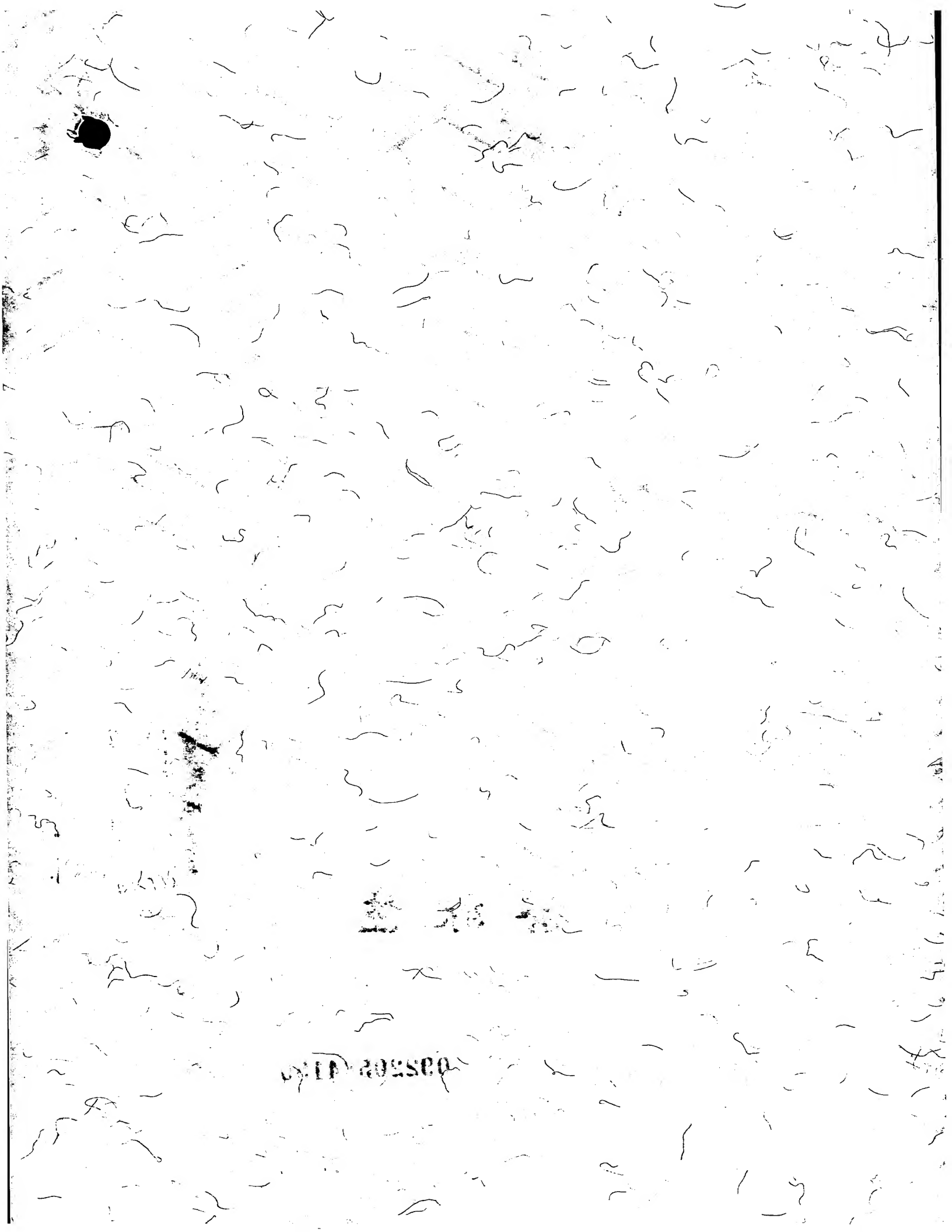
申請人：國立清華大學
Applicant(s)

局長
Director General

蔡練生

發文日期：西元 2003 年 6 月 13 日
Issue Date

發文字號：09220584150
Serial No.



0510-20580

發明專利說明書

(填寫本書件時請先行詳閱申請書後之申請須知，作※記號部分請勿填寫)

※ 申請案號：_____ ※IPC分類：_____

※ 申請日期：_____

壹、發明名稱

(中文) 多埠記憶體之測試方法

(英文) Method of Multi-port Memory Test

貳、發明人(共 4 人)

發明人 1 (如發明人超過一人，請填說明書發明人續頁)

姓名：(中文) 吳誠文

(英文) Cheng-Wen Wu

住居所地址：(中文) 300 新竹市光復路二段 101 號

(英文)

國籍：(中文) 中華民國

(英文)

參、申請人(共 1 人)

申請人 1 (如申請人超過一人，請填說明書申請人續頁)

姓名或名稱：(中文) 國立清華大學

(英文) National Tsing Hua University

住居所或營業所地址：(中文) 300 新竹市光復路二段 101 號

(英文)

國籍：(中文) 中華民國

(英文)

代表人：(中文) 徐遐生

(英文) Hsia-San Shu

☒ 續發明人或申請人續頁 (發明人或申請人欄位不敷使用時，請註記並使用續頁)

發明人 2

姓名：(中文) 黃稚存
(英文) Chih-Tsun Huang

住居所地址：(中文) 300 新竹市光復路二段 101 號
(英文)

國籍：(中文) 中華民國 (英文)

發明人 3

姓名：(中文) 王志偉
(英文) Chih-Wea Wang

住居所地址：(中文) 300 新竹市光復路二段 101 號
(英文)

國籍：(中文) 中華民國 (英文)

發明人 4

姓名：(中文) 鄭國良
(英文) Kuo-Liang Cheng

住居所地址：(中文) 300 新竹市光復路二段 101 號
(英文)

國籍：(中文) 中華民國 (英文)

發明人 5

姓名：(中文)
(英文)

住居所地址：(中文)
(英文)

國籍：(中文) (英文)

發明人 6

姓名：(中文)
(英文)

住居所地址：(中文)
(英文)

國籍：(中文) (英文)

肆、中文發明摘要

一種多埠記憶體之測試方法，該多埠記憶體具有複數存取埠可平行存取其所有之記憶晶胞(記憶位址)，且該測試方法包含至少一測試單元，而該測試單元包含至少一測試動作，並藉由該等存取埠依序對每一記憶晶胞(記憶位址)進行所包含之測試動作，特別是，該測試方法更在該測試單元包含複數測試動作時，將該等測試動作分派給該等存取埠，並使該等存取埠在一記憶晶胞(記憶位址)之測試時脈週期中的不同時點對同一記憶晶胞進行被分派之測試動作，藉此有效縮短多埠記憶體之測試時間。

伍、英文發明摘要

(一)、本案指定代表圖為：第四圖

(二)、本代表圖之元件代表符號簡單說明：

(無)

柒、本案若有化學式時，請揭示最能顯示發明特徵的化學式：_____

捌、聲明事項

☐ 本案係符合專利法第二十條第一項第一款但書或第二款但書規定之期間，其日期為：_____

☐ 本案已向下列國家（地區）申請專利，申請日期及案號資料如下：

【格式請依：申請國家（地區）；申請日期；申請案號 順序註記】

1. _____
2. _____
3. _____

☐ 主張專利法第三十四條第一項優先權

【格式請依：受理國家（地區）；日期；案號 順序註記】

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
9. _____
10. _____

☐ 主張專利法第三十五條之一第一項優先權

【格式請依：申請日；申請案號 順序註記】

1. _____
2. _____
3. _____

☐ 主張專利法第三十六條微生物

☐ 國內微生物 【格式請依：寄存機構；日期；號碼 順序註記】

1. _____
2. _____
3. _____

☐ 國外微生物 【格式請依：寄存國名；機構；日期；號碼 順序註記】

1. _____
2. _____
3. _____

☐ 熟習該項技術者易於獲得，不須寄存。

【發明所屬之技術領域】

本發明是有關於一種多埠記憶體之測試方法，特別是指一種可有效縮短測試時間之多埠記憶體之測試方法。

【先前技術】

5 在 SOC(System-On-Chip,系統單晶片)產品中，將數百
個記憶體磁芯集積在單一晶片中的情況很普遍，且在 SOC
晶片中內建的記憶體區塊佔用的面積逐漸增加，近幾年平
均更將高達 90%。而除了記憶容量，SOC 晶片資料頻寬
10 的需求也迅速成長，因此，具有較長字組及較多存取埠
(access port)的多埠記憶體被開發出來。此外，多埠記憶
體亦被廣泛使用在多處理器系統、網路處理器、繪圖晶片
和其他高效能裝置上，同時，它們也被用來支援不同時脈
區域之間的資料通訊。因此，隨著多埠記憶體的需求增加
，如何在產品開發階段有效且迅速地偵測及診斷多埠記憶
15 體中的缺陷，以及在量產階段作有效率的測試，遂成為一
重要的技術課題。

然而，與傳統單埠記憶體不同的是，多埠記憶體提供
一平行存取通道，其可由不同的埠去存取每一(或同一)記
憶晶胞，所以多埠記憶體和單埠記憶體在硬體架構上是很
20 不一樣的，因為大部分傳統單埠記憶體的記憶晶胞被定義
成與晶胞結構和陣列幾何是獨立的(不相關的)，但是，多
埠記憶體的埠內部損壞卻是結構相依的，亦即其損壞行為
和電路設計及處理技術是高度相依的，因此，當多埠記憶
體在相鄰的不同埠之位元線之間或字元線之間有短路產生

時，這些損壞將比傳統單埠記憶體之損壞更難被偵測出來。因此，為了提升多埠記憶體之損壞偵測效能，近年來有許多針對多埠記憶體之測試方法被提出，包括 Zero-One、Checkerboard、GALPAT、Walking 1/0、Sliding Diagonal、Butterfly 和 March 演算法等等。其中又以 March 演算法較其他演算法之測試效能來得優異。而且 March 演算法針對記憶體中不同的錯誤模型(fault model)更衍生出 MATS+、Marching 1/0、MATS++、March X、March C+/C-、March A、March Y 及 March B 等演算法。

以 March C-演算法為例，其測試式子為 $\{\updownarrow(wa); \uparrow(ra, wb); \uparrow(rb, wa); \downarrow(ra, wb); \downarrow(rb, wa); \updownarrow(ra)\}$ ，其中， $a=0$ 或 1 ， $b=\bar{a}$ ，箭頭 \updownarrow 代表測試之位址順序由上往下或由下往上皆可，箭頭 \uparrow 代表測試之位址順序是由下往上，而箭頭 \downarrow 代表測試之位址順序是由上往下，且每一括弧 $()$ 為一個測試單元(test element)。每一測試單元中包含讀 $a(ra)$ 、寫 $b(wb)$ 、讀 $b(rb)$ 、寫 $a(wa)$ 等測試動作之至少其中之一或其連續動作。且多埠記憶體中之每一記憶晶胞(或記憶位址)必需完成一測試單元中之所有測試動作後，才能進行下一測試單元。如第一圖所示，假設多埠記憶體 1 具有兩個存取埠 PORT A 和 PORT B，當進行例如

$\uparrow(ra, wb)$ 測試單元時，習知的一種作法是，如第二圖所示，由 PORT A 及 PORT B 兩者輪流對每一記憶晶胞(或記憶位址)做 ra 和 wb 之測試動作，或者，如第三圖所示，由 PORT A 及 PORT B 先後對所有記憶晶胞(或記憶位址)做

↑(ra,wb)測試。因此，如第二圖或第三圖所示，不論是
PORT A 及 PORT B 輪流測試或先後測試，因為當其中一
PORT 動作時，另一 PORT 即不動作，以致每一測試動作
(ra 或 wb)都需花費一個測試時脈週期，所以，假設 N 是
5 記憶體陣列的位元數，則由計算可以得知傳統測試方法的
時間複雜度(測試時間)為 $10N$ 。而且由於多埠記憶體的錯
誤型態繁雜，常常需要使用複雜的演算法來進行偵測，使
得當多埠記憶體的容量越來越大且記憶型態越驅複雜，導
致演算法越來越繁瑣時，將使測試時間大量增加，而嚴重
10 影響多埠記憶體的測試效能。

【發明內容】

因此，本發明之目的，在於利用多埠記憶體之不同埠
在同一存取時脈中的不同時點對同一記憶晶胞進行測試動
作，而達到有效縮短測試時間之多埠記憶體之測試方法。

15 於是，本發明多埠記憶體之測試方法，該多埠記憶體
具有複數存取埠及可供該等存取埠存取之複數記憶晶胞，
該測試方法包含至少一測試單元，該測試單元包含讀 1、
寫 0、讀 0、寫 1 等測試動作的至少其中之一，且該測試
單元係藉由該等存取埠依序對每一記憶晶胞進行所包含之
20 測試動作，特別是，該測試方法更包含下列步驟：當該測
試單元包含複數測試動作時，將該等測試動作分派給該等
存取埠，並使該等存取埠在一記憶晶胞之測試時脈中的不
同時點對同一記憶晶胞進行分派之測試動作。藉此，達到
有效縮短多埠記憶體的測試時間之功效。

再者，本發明多埠記憶體之測試方法，該多埠記憶體具有複數存取埠以及可供該等存取埠存取之複數記憶位址，該測試方法包含至少一測試單元，該測試單元包含至少一測試動作，且該測試單元係藉由該等存取埠依序對所有記憶位址進行所包含之測試動作，特別是，該測試方法更包含下列步驟：當該測試單元包含複數測試動作時，將該等測試動作分派給該等存取埠，並使該等存取埠在一記憶位址之測試時脈中的不同時點對同一記憶位址進行分派之測試動作。藉此，達到有效縮短多埠記憶體的測試時間之功效。

此外，本發明之電腦程式產品，用以執行多埠記憶體之測試，該多埠記憶體具有複數存取埠及可供該等存取埠存取之複數記憶晶胞，該電腦程式包含至少一測試單元，該測試單元包含讀 1、寫 0、讀 0、寫 1 等測試動作的至少其中之一，且該測試單元係藉由該等存取埠依序對每一記憶晶胞進行所包含之測試動作，特別是，該電腦程式更包含下列步驟：當該測試單元包含複數測試動作時，將該等測試動作分派給該等存取埠，並使該等存取埠在一記憶晶胞之測試時脈中的不同時點對同一記憶晶胞進行分派之測試動作。

另外，本發明之電腦程式產品，用以執行多埠記憶體之測試，該多埠記憶體具有複數存取埠及可供該等存取埠存取之複數記憶晶胞，該電腦程式包含至少一測試單元，該測試單元包含讀 1、寫 0、讀 0、寫 1 等測試動作的至

少其中之一，且該測試單元係藉由該等存取埠依序對每一記憶晶胞進行所包含之測試動作，特別是，該電腦程式更包含下列步驟：當該測試單元包含複數測試動作時，將該等測試動作分派給該等存取埠，並使該等存取埠在一記憶晶胞之測試時脈中的不同時點對同一記憶晶胞進行分派之測試動作。

【實施方式】

本發明之前述以及其他技術內容、特點與功效，在以下配合參考圖式之數較佳實施例的詳細說明中，將可清楚的明白。

本發明多埠記憶體之測試方法的第一較佳實施例仍以上述 March C-演算法為例，以便與前述之傳統測試方法做一比較。且本發明之測試方法所測試的對象亦是一具有雙存取埠 PORT A 和 PORT B 之多埠記憶體(如第一圖所示)。而且，同樣以測試單元 $\uparrow(ra, wb)$ 為例，當對所有記憶晶胞(或記憶位址 $k, k+1, k+2, \dots$)進行 $\uparrow(ra, wb)$ 測試時，本發明的特徵在於：如第四圖所示，當測試單元中的測試動作為複數時，即將該等測試動作適當分派給所有存取埠，因此，測試單元 $\uparrow(ra, wb)$ 中的測試動作 (ra, wb) 被分派給 PORT A(執行 ra)及 PORT B(執行 wb)，並使 PORT A 和 PORT B 在一記憶晶胞(記憶位址)之測試時脈 clk 中的不同時點，對同一記憶晶胞(記憶位址 $K, k+1, k+2, \dots$)進行被分派之測試動作，亦即 $(\overset{ra}{wb})$ ，且為確保在同一測試時脈週期中的下一測試動作(wb)是在前一測試動作(ra)執行

後才開始，兩測試動作 (\uparrow_{wb}^{ra}) 之間具有一時間差 t_{cc} ，且時間差 $t_{cc} \leq 1/2 t_{cyc}$ (測試時脈週期)。因此，藉由上述測試方法，可將原本的 March C-演算法 $\{\uparrow_{wb}^{ra}; \uparrow(r_a, w_b); \uparrow(r_b, w_a); \downarrow(r_a, w_b); \downarrow(r_b, w_a); \uparrow(r_a)\}$ 改變成如下式子：

5 $\{\uparrow_{wb}^{ra}; \uparrow_{wb}^{ra}; \uparrow_{wb}^{rb}; \downarrow_{wb}^{ra}; \downarrow_{wb}^{rb}; \uparrow(r_a)\}$ ，其中，(\uparrow_{wb}^{rb}) 表示經由兩個不同的存取埠 PORT A 及 PORT B 在同一個測試時脈週期中對同一記憶晶胞進行 r_b 及 w_a 測試 (此一技術亦被稱之為 sequence folding (序列疊合技術))。藉此，可將測試時間複雜度由原來的 $10N$ 縮短為

10 $6N$ (N 為記憶體陣列的位元數)。

而且，當被測試之多埠記憶體的存取埠越多時，應用本發明之測試方法更可進一步縮短測試時間。如第五圖所示，以 Extended March C-演算法為例，其演算法如下：

15 $\{\uparrow_{wb}^{ra}; \uparrow(r_a, w_b); \uparrow(r_b, w_a); \downarrow(r_a, w_b, r_b); \downarrow(r_b, w_a, r_a); \uparrow(r_a)\}$

首先考慮被測對象為一單埠記憶體的情況，或是一多埠記憶體但是以多埠輪流或單埠依序對所有記憶晶胞 (記憶位址) 進行測試的情況時，其測試的時間複雜度為 $12N$ (N 是記憶體的位元數)。而當被測對象是一具有雙存取埠 PORT A 及 PORT B 的多埠記憶體，並應用本發明之測試方法 --- sequence folding (序列疊合技術) 時，即可將測試單元中包含兩個以上的測試動作適當分派給存取埠 PORT A 及 PORT B，使在同一測試時脈中對同一記憶晶胞 (記憶位址) 同時進行二個測試動作，例如，將原先測試

20

每一記憶晶胞(記憶位址)需要兩個測試時脈的測試單元
 $\downarrow (ra, wb)$ 變成測試每一記憶晶胞(記憶位址)僅需 1 個測試時脈即可完成之測試單元 $\downarrow \begin{matrix} ra \\ wb \end{matrix}$ ，藉此，可將測試時間(時間複雜度)由原來的 $12N$ 降至 $8N$ 。再者，當被測對象是一具有三個存取埠 PORT A~C 時，應用本發明之測試方法---sequence folding(序列疊合技術)，更可將包含有三個測試動作的測試單元，例如 $\downarrow (ra, wb, rb)$ 中的所有測試動作平均分派給三個存取埠 PORT A~C，則如第六圖所示之存取埠 PORT A~C 的測試動作，以及第七圖所示之存取埠 PORT A~C 的時脈所示，將原先測試一個記憶晶胞(記憶位址)需要三個測試時脈的測試單元 $\downarrow (ra, wb, rb)$ 變成測試一記憶晶胞(記憶位址)僅需 1 個測試時脈即可完成之測試單元 $\downarrow \begin{matrix} ra \\ wb \\ rb \end{matrix}$ ，使存取埠 PORT A~C 在同一測試時脈中對同一記憶晶胞進行測試動作 (wb)，藉而進一步將測試時間(時間複雜度)由 $8N$ 縮短至 $6N$ 。而且，如第六、七圖所示，本發明之測試方法在前後兩測試動作之間具有一時間差 t_{cc} ，且三個測試動作的執行時間總和需小於(或等於)一個測試時脈週期 t_{cyc} 。

接著，如第八~十二圖所示，是本發明多埠記憶體之測試方法的第二較佳實施例。本實施例主要是結合一 port scheduling(埠排程法)技術使用。首先，要先說明的是，port scheduling 技術是一種藉由適當安排演算法中的測試單元給多埠記憶體的各個存取埠，以達到縮短排程及測試時間並提高存取埠使用率等目的之方法。舉一例

子說明，第八圖所示是一完整的測試演算法，其用以偵測一具有雙埠 PORT A 及 PORT B 之多埠記憶體的所有內部錯誤，包括所有 SAF(stuck-at fault)、TF(transition fault)、SOF(stuck-open fault)、DRF(data retention fault)和 CF(coupling fault)等損壞，且該演算法的全部測試單元被區分成三個測試區塊(區塊 1~區塊 3)，每個區塊 1~3 針對不同的損壞錯誤，例如區塊 1 是針對記憶體陣列錯誤，區塊 2 是針對定址錯誤，區塊 3 是針對埠或埠之間的存取錯誤等進行偵測。其中，如第八圖所示，大部分的測試動作都是經由 PORT A 執行，且當 PORT B 要與 PORT A 存取同一位址時，也是在 PORT A 不動作時才進行，要不就是在 PORT A 對某一記憶晶胞(記憶位址)進行測試時，由 PORT B 對與該記憶晶胞(記憶位址)相鄰的其他記憶晶胞(記憶位址)進行測試，因此，完成整個測試的時間複雜度為 $24N$ (N 是記憶體陣列的位元數)。但是，由第八圖中顯示可知，在區塊 2 和區塊 3 中，有部分的測試單元和區塊 1 中的測試單元重覆，因此，藉由 port scheduling(埠排程法)，將第八圖之演算法經由第九~十一圖進行一連串的行程編排後，將區塊 2 與區塊 3 適當併入區塊 1 中，在測試效果不變的情況下，簡化了測試行程，並且將測試時間(時間複雜度)由原來的 $24N$ 縮短至 $10N$ ，而且提高 PORT B 的使用率。可是，由於此一埠排程法所編排之演算法，仍是只能在單一測試時脈週期中執行一測試動作，因此，如第十一圖所示，當某一存取埠進行測

試單元中的測試動作時，另一存取埠則仍處於閒置狀態，而未被善加利用。因此，如第十二圖所示，利用本發明之測試方法---sequence folding(序列疊合技術)，再對經過 port scheduling(埠排程法)編排過之演算法進行 sequence folding (序列疊合)，將 $\downarrow(r0w1)$ 及 $\downarrow(r1w0)$ 分別變成 $\downarrow(\overset{r0}{w1})$ 及 $\downarrow(\overset{r1}{w0})$ ，並使 PORT A 和 PORT B 在同一時脈中對同一記憶晶胞進行 $(\overset{r0}{w1})$ 及 $(\overset{r1}{w0})$ 測試，而更進一步地將測試時間由 10N 縮短為 8N。

此外，值得一提的是，本發明多埠記憶體之測試方法可由一電子裝置所執行，且該電子裝置中載有本發明用以控制該電子裝置執行多埠記憶體測試之電腦程式。

當然，本發明之測試方法並不僅限於應用在上述實施例所揭露之 March 演算法，也就是說，其他在其測試單元中包含有複數測試動作之演算法，諸如 MSCAN 及 Butterfly 等測試演算法皆可應用本發明之測試方法來縮短其測試時間。

綜上所述，本發明主要是針對多埠記憶體之測試演算法，將演算法之測試單元中包含兩個以上的測試動作分派給多埠記憶體的所有存取埠執行，使在同一測試時脈中對同一記憶晶胞(記憶位址)執行不同之測試動作，藉而有效縮短多埠記憶體之測試時間，並使多埠記憶體之存取埠的利用率達到百分之百，確實達到本發明之目的。

惟以上所述者，僅為本發明之較佳實施例而已，當不能以此限定本發明實施之範圍，即大凡依本發明申請專利

範圍及發明說明書內容所作之簡單的等效變化與修飾，皆應仍屬本發明專利涵蓋之範圍內。

【圖式簡單說明】

第一圖是一雙埠記憶體之硬體架構方塊示意圖；

5 第二圖是習知一種以雙埠記憶體之雙埠輪流測試記憶晶胞(記憶位址)之測試方法的時序圖；

第三圖是習知一種以雙埠記憶體之單埠測試記憶晶胞(記憶位址)之測試方法的時序圖；

10 第四圖是本發明多埠記憶體之測試方法的第一較佳實施例，其中顯示在一測試時脈週期中，以一雙埠記憶體之雙埠同時測試記憶晶胞(記憶位址)之時序圖；

第五圖是第一實施例的另一實施態樣，其中顯示在單一測試時脈中以單埠、雙埠及三埠同時對同一記憶晶胞(記憶位址)進行測試動作所花費之測試時間比較結果；

15 第六圖顯示三埠記憶體的存取埠 PORT A~C 之測試動作時序；

第七圖顯示三埠記憶體之存取埠 PORT A~C 的時脈之間的時間差。

20 第八圖~第十一圖用以說明一測試演算法經過 port scheduling(埠排程法)之編排過程；及

第十二圖顯示第十一圖之演算法中的測試單元經過本發明之測試方法重新編排後之結果。

【圖式之主要元件代表符號簡單說明】

1 雙埠記憶體

1. 一種多埠記憶體之測試方法，該多埠記憶體具有複數存取埠及可供該等存取埠存取之複數記憶晶胞，該測試方法包含至少一測試單元，該測試單元包含讀 1、寫 0、讀 0、寫 1 等測試動作的至少其中之一，且該測試單元係藉由該等存取埠依序對每一記憶晶胞進行所包含之測試動作，其特徵在於：

該測試方法更包含下列步驟：

當該測試單元包含複數測試動作時，將該等測試動作分派給該等存取埠，並使該等存取埠在一記憶晶胞之測試時脈中的不同時點對同一記憶晶胞進行被分派之測試動作。

2. 依申請專利範圍第 1 項所述多埠記憶體之測試方法，其中，該等測試動作之間具有一時間差，且該等時間差總和不超過該測試時脈週期。
3. 依申請專利範圍第 1 項所述多埠記憶體之測試方法，其中該測試方法包含複數測試單元，且該等測試單元是由 March 演算法所產生。
4. 一種多埠記憶體之測試方法，該多埠記憶體具有複數存取埠以及可供該等存取埠存取之複數記憶位址，該測試方法包含至少一測試單元，該測試單元包含至少一測試動作，且該測試單元係藉由該等存取埠依序對所有記憶位址進行所包含之測試動作，其特徵在於：

該方法更包含下列步驟：

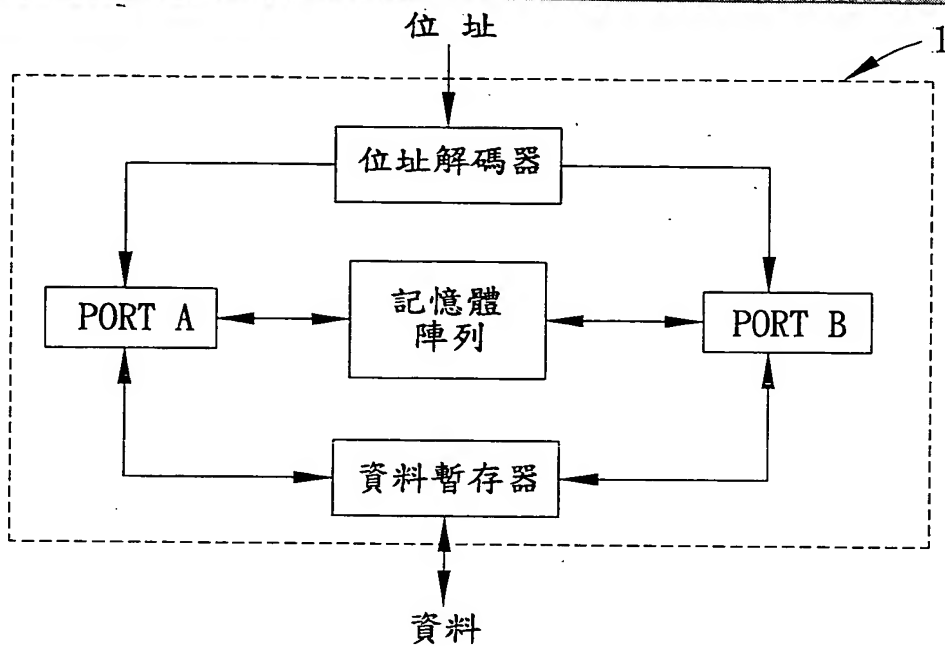
當該測試單元包含複數測試動作時，將該等測試動作

分派給該等存取埠，並使該等存取埠在一記憶位址之測試時脈中的不同時點對同一記憶位址進行被分派之測試動作。

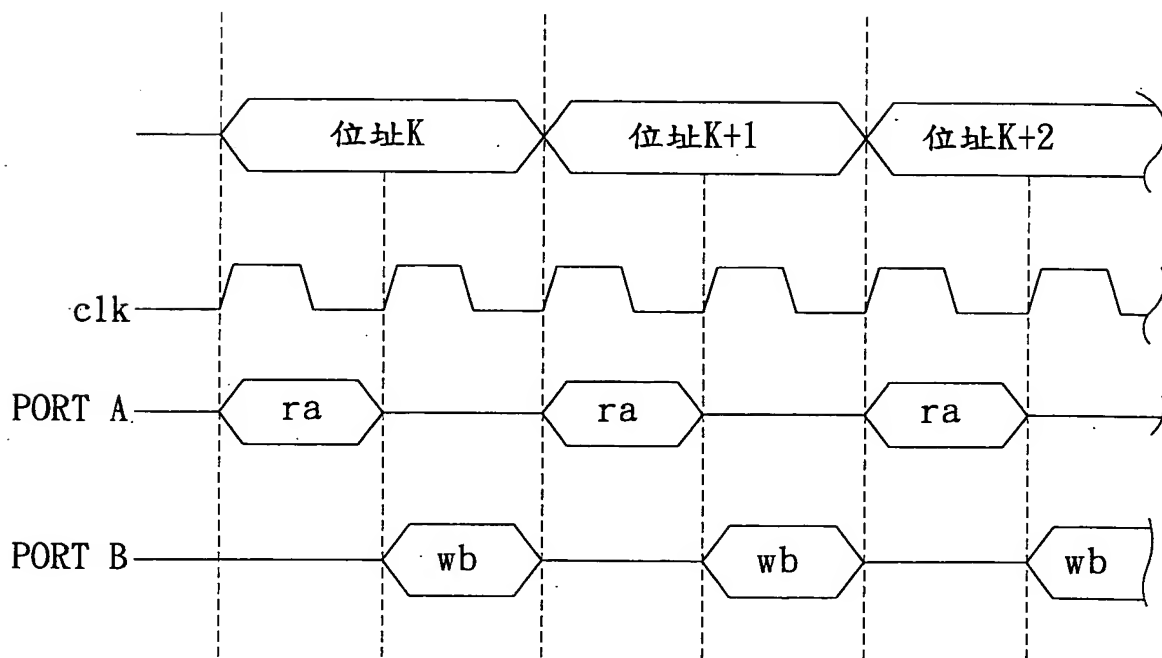
5. 依申請專利範圍第4項所述多埠記憶體之測試方法，其中該等測試動作之間具有一時間差，且該等時間差總和不超過該測試時脈週期。
6. 依申請專利範圍第4項所述多埠記憶體之測試方法，其中該測試單元包含之測試動作為讀1、寫0、讀0、寫1等動作的至少其中之一。
7. 依申請專利範圍第4項所述多埠記憶體之測試方法，其中該測試方法包含複數測試單元，且該等測試單元是由March演算法所產生。
8. 一種電腦程式產品，用以執行多埠記憶體之測試，該多埠記憶體具有複數存取埠及可供該等存取埠存取之複數記憶晶胞，該電腦程式包含至少一測試單元，該測試單元包含讀1、寫0、讀0、寫1等測試動作的至少其中之一，且該測試單元係藉由該等存取埠依序對每一記憶晶胞進行所包含之測試動作，其特徵在於：
該電腦程式更包含下列步驟：
當該測試單元包含複數測試動作時，將該等測試動作分派給該等存取埠，並使該等存取埠在一記憶晶胞之測試時脈中的不同時點對同一記憶晶胞進行被分派之測試動作。
9. 依申請專利範圍第8項所述之電腦程式產品，其中，該等測

試動作之間具有一時間差，且該等時間差總和不超過該測試時脈週期。

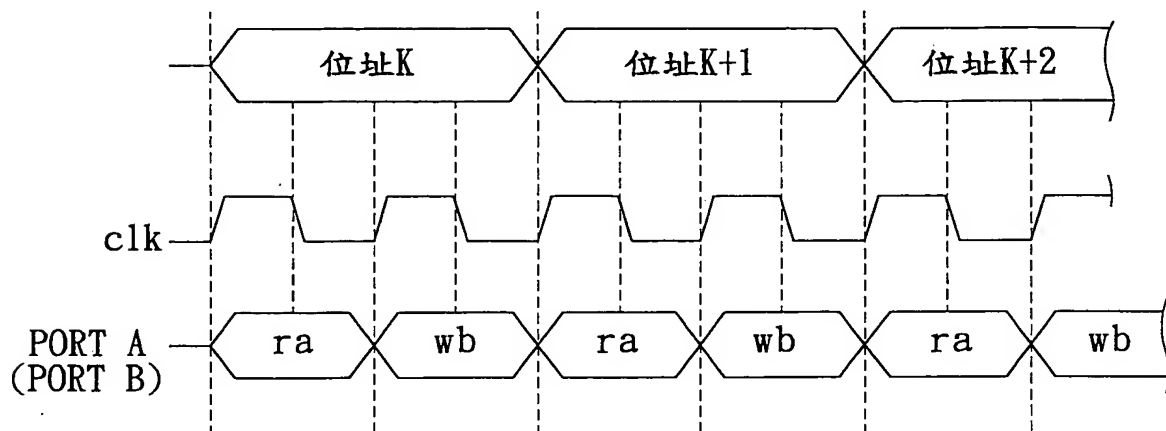
10. 依申請專利範圍第8項所述之電腦程式產品，其中該電腦程式包含複數測試單元，且該等測試單元是由March演算法所產生。
11. 一種電腦程式產品，用以執行多埠記憶體之測試，該多埠記憶體具有複數存取埠以及可供該等存取埠存取之複數記憶位址，該電腦程式包含至少一測試單元，該測試單元包含至少一測試動作，且該測試單元係藉由該等存取埠依序對所有記憶位址進行所包含之測試動作，其特徵在於：
該電腦程式更包含下列步驟：
當該測試單元包含複數測試動作時，將該等測試動作分派給該等存取埠，並使該等存取埠在一記憶位址之測試時脈中的不同時點對同一記憶位址進行被分派之測試動作。
12. 依申請專利範圍第11項所述之電腦程式產品，其中該等測試動作之間具有一時間差，且該等時間差總和不超過該測試時脈週期。
13. 依申請專利範圍第11項所述之電腦程式產品，其中該測試單元包含之測試動作為讀1、寫0、讀0、寫1等動作的至少其中之一。
14. 依申請專利範圍第11項所述之電腦程式產品，其中該電腦程式包含複數測試單元，且該等測試單元是由March演算法所產生。



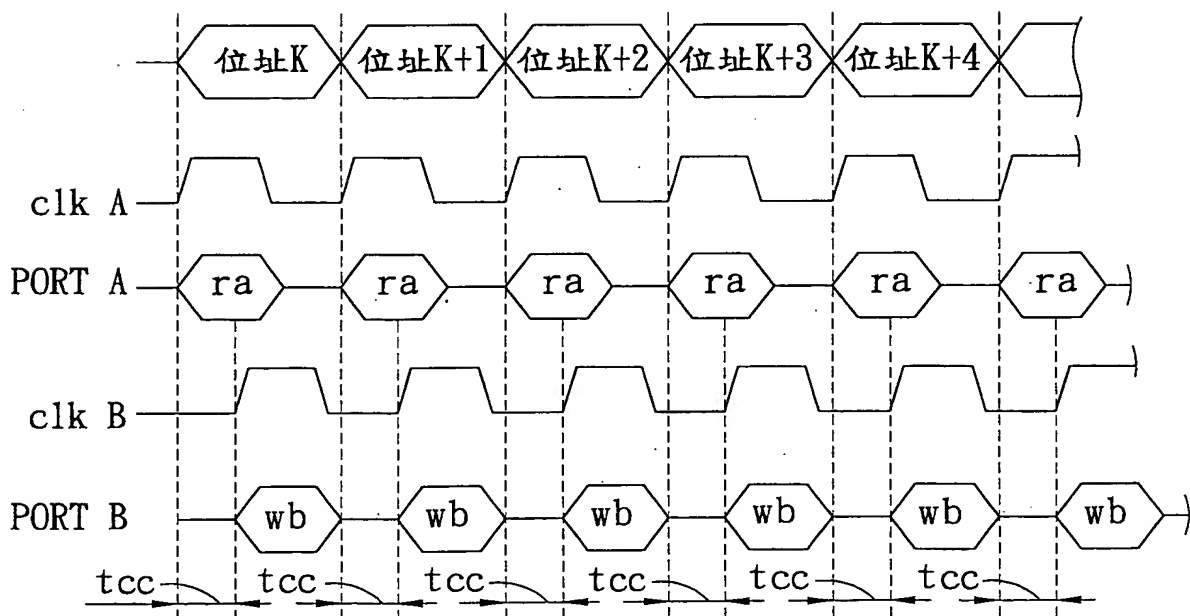
第一圖



第二圖



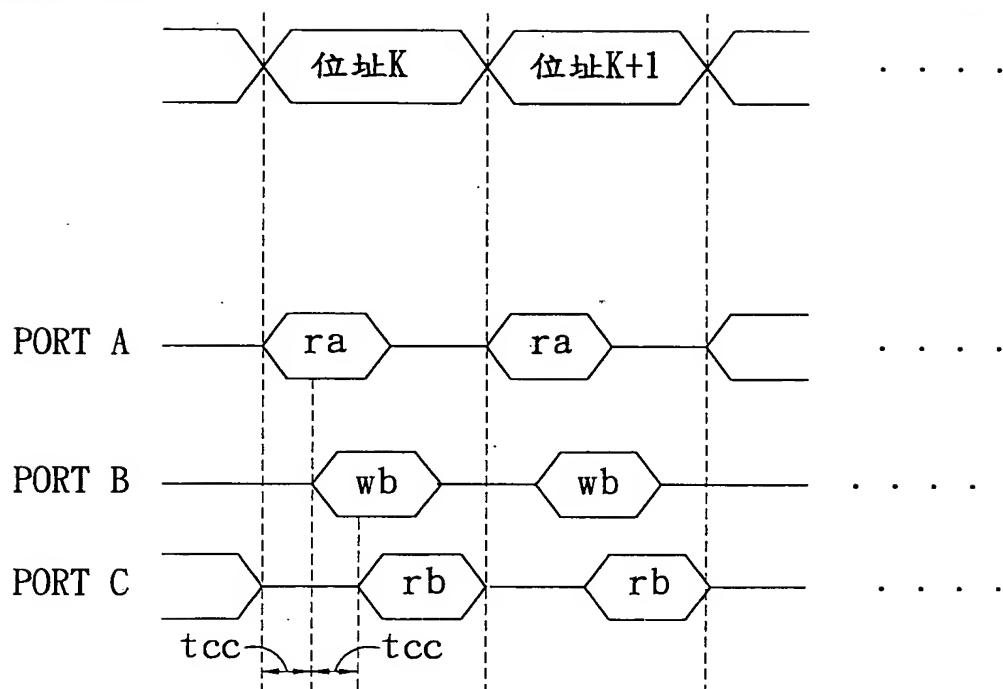
第三圖



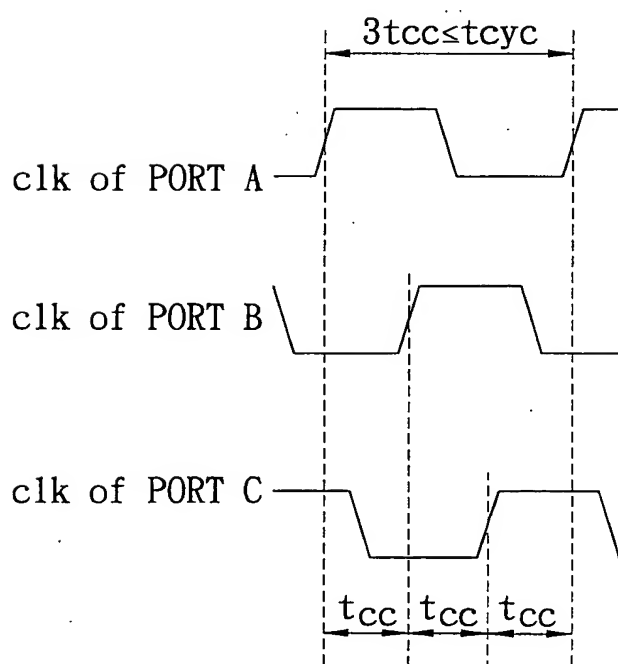
第四圖

| Extended March C -Test | | 測試 時間 |
|------------------------|---|----------|
| 單埠 | PA: $\Downarrow(wa) \Uparrow(ra, wb) \Uparrow(rb, wa) \Downarrow(ra, wb, rb) \Downarrow(rb, wa, ra) \Uparrow(ra)$ | 12N |
| 雙埠 | PA: $\Downarrow(wa) \Uparrow(ra) \Uparrow(rb) \Downarrow(ra, rb) \Downarrow(rb, ra) \Uparrow(ra)$ PB: $(-)(-)(wb)(wa)(wb, -)(wa, -)(-)$ | 8N |
| 三埠 | PA: $\Downarrow(wa) \Uparrow(ra) \Uparrow(rb) \Downarrow(ra) \Downarrow(rb) \Uparrow(ra)$ PB: $(-)(-)(wb)(wa)(wb)(wa)(-)$ PC: $(-)(-)(-)(-)(rb)(ra)(-)$ | 6N |

第五圖



第六圖



第七圖

